Recurso de Algoritmos e Estruturas de Dados II

Departamento de Informática

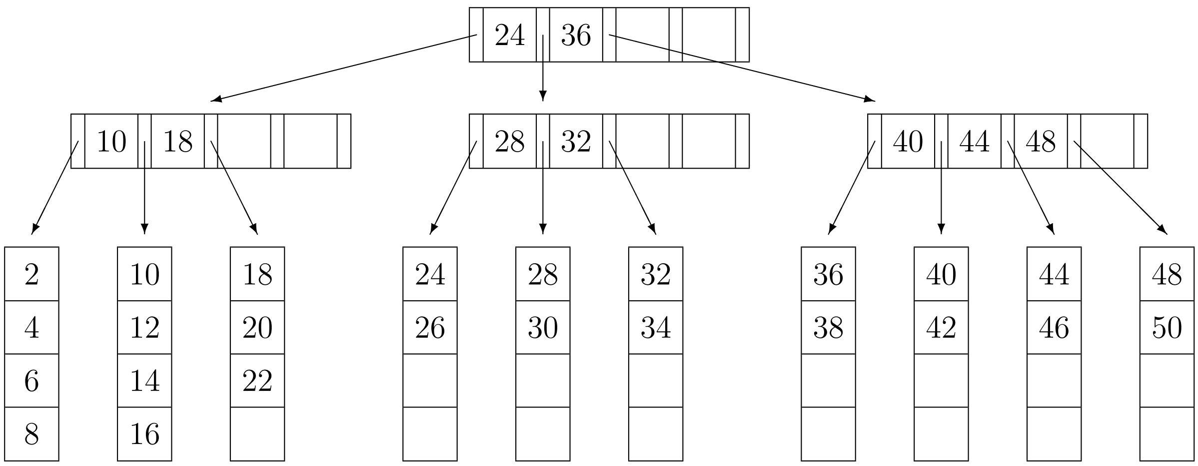
Universidade Nova de Lisboa

8 de Fevereiro de 2007

Por favor, entregue as respostas em folhas distintas.

1. (a) [1.5 valores] Calcule a ordem de uma árvore B^+ cujas chaves ocupam 16 bytes, assumindo que cada página ocupa um bloco de 4096 bytes e que cada apontador para uma página ocupa 4 bytes. Justifique todos os cálculos que efectuar.

   (b) [2.5 valores] Considere a árvore B^+ de ordem 5 e capacidade 4 esquematizada na figura. Apresente uma representação gráfica da árvore após a realização sucessiva de cada uma das quatro seguintes operações: Inserir 15, Inserir 3, Remover 36 e Remover 34.



2. [3 valores] Sejam $X = < x_1\, x_2 \cdots x_n >$ e $Y = < y_1\, y_2 \cdots y_n >$ duas sequências não vazias de zeros, uns e dois. Diz-se que X e Y *colidem* se, e só se, X e Y possuem o mesmo elemento numa (qualquer) mesma posição:

$$(\exists i \in \{1, 2, \ldots, n\}) \;\; x_i = y_i \, .$$

Seja, agora, $C = \{S_1, S_2, \ldots, S_k\}$ um conjunto não vazio de sequências de zeros, uns e dois, todas de comprimento n. A sequência X *cobre* C se, e só se, X colide com todas as sequências de C:

$$(\forall S \in C) \;\; X \text{ e } S \text{ colidem.}$$

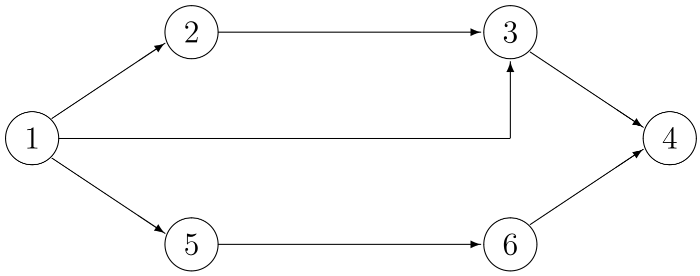
Por exemplo, a sequência $< 0\,1\,0 >$ cobre o conjunto

$$\{< 0\,0\,2 >,\ < 0\,1\,2 >,\ < 0\,2\,1 >,\ < 0\,2\,2 >,\ < 1\,1\,0 >,\ < 1\,2\,0 >,\ < 2\,1\,2 >,\ < 2\,2\,0 >\}.$$

O **Problema da Cobertura de Sequências** formula-se da seguinte forma.
Dado um conjunto não vazio, $C = \{S_1, S_2, \ldots, S_k\}$, de sequências de zeros, uns e dois, todas de comprimento positivo n, existe uma sequência X, de zeros e uns e de comprimento n que cobre C?

Prove que o Problema da Cobertura de Sequências é **NP**.

3. [3 valores] Suponha que se executa o algoritmo **ordenaçãoTopológica** com o grafo esquematizado na figura.

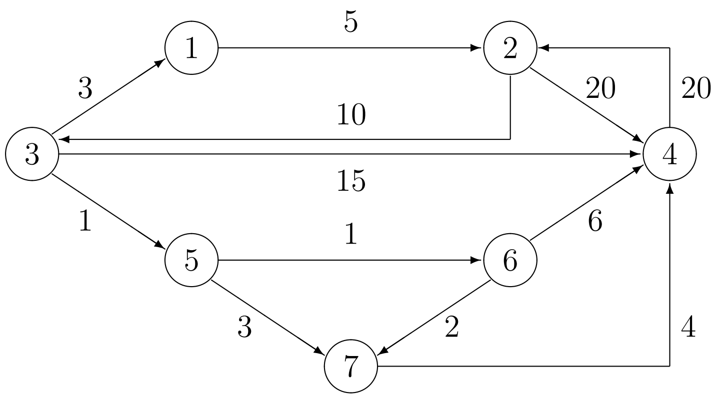


Para simplificar, assuma que a expressão *TRATAR(vertice)* apenas imprime o vértice dado. Indique a saída do algoritmo (ou seja, a ordem pela qual os vértices são tratados) se, quando se itera o conjunto dos sucessores do vértice 1:

(a) [1.5 valores] a iteração produz **2**, **3** e **5** (primeiro o vértice 2, depois o vértice 3 e, por fim, o vértice 5);

(b) [1.5 valores] a iteração produz **5**, **3** e **2** (primeiro o vértice 5, depois o vértice 3 e, por fim, o vértice 2).

4. [4 valores] Seja G um grafo orientado e pesado, cujos arcos têm custo positivo. Dados dois quaisquer vértices, v e w, pretende-se determinar o número de caminhos de v para w cujo comprimento (pesado) é mínimo.



Por exemplo, no grafo representado na figura, há **três** caminhos do vértice 3 para o vértice 4 cujo comprimento (pesado) é mínimo.

Comprimento Pesado	**Caminho**
8	3 5 7 4
8	3 5 6 4
8	3 5 6 7 4

Apresente um algoritmo (em pseudo-código) que, dados um grafo $G = (V, A)$ orientado e pesado, cujos arcos têm custo positivo, e dois vértices, v e w, determine o número de caminhos de v para w cujo comprimento (pesado) é mínimo. Para simplificar, pode assumir que há, pelo menos, um caminho entre v e w. Estude a complexidade temporal do seu algoritmo, no pior caso.

5. [3 valores] Considere a seguinte função recursiva $F(i, j)$, onde i e j são inteiros não negativos.

$$F(i,j) = \begin{cases} j, & \text{se } i = 0 \text{ e } j \geq 0; \\ F(i-1,j)^2, & \text{se } i = 1 \text{ e } j \geq 0; \\ 5i, & \text{se } i \geq 2 \text{ e } j = 0; \\ \max(\ F(i-2,j) + F(i-1,j-1), & \\ \qquad F(i-2,j-1) + F(i-1,j)\ ), & \text{se } i \geq 2 \text{ e } j \geq 1. \end{cases}$$

Apresente um algoritmo, desenhado segundo a técnica da programação dinâmica, que, dados dois inteiros positivos M e N, calcula o valor da função $F(M, N)$. Estude a complexidade temporal e espacial do seu algoritmo, no pior caso.

6. [3 valores] A sua área pessoal está muito desorganizada: contém imensos ficheiros na raiz que devem ser movidos para diversas directorias. Para a organizar, pode seleccionar o primeiro ficheiro e movê-lo para a directoria respectiva, depois, seleccionar o segundo ficheiro e movê-lo para a directoria correspondente, e assim sucessivamente, efectuando tantas selecções quanto o número de ficheiros. Seleccionando um conjunto de ficheiros contíguos cujo destino é o mesmo e movendo-os de uma só vez, poderá diminuir o número total de operações de selecção. Depois de mover um ou mais ficheiros, estes desaparecem da directoria raiz, e outros ficheiros com a mesma directoria de destino poderão tornar-se adjacentes.

   Pretende-se determinar o número mínimo de operações de selecção necessárias para mover todos os ficheiros para as respectivas directorias.

   Por exemplo se as directorias de destino forem

   `A, B, C, B, A,`

   poderemos mover o terceiro ficheiro para a directoria `C`, em seguida, os dois ficheiros que têm como destino a directoria `B` e, finalmente, os ficheiros para a directoria `A`. Procedendo desta forma, efectuam-se três selecções, sendo este o número mínimo de operações de selecção necessárias para mover os cinco ficheiros para as respectivas directorias.

   Se as directorias de destino fossem `A`, `B`, `C`, `B`, `B`, `A`, `C`, seriam necessárias quatro operações de selecção.

   Apresente **uma função recursiva** que, dada uma sequência D (de comprimento $n \geq 2$) com os nomes das directorias de destino dos ficheiros, calcula o número mínimo de operações de selecção necessárias para mover todos os ficheiros para as respectivas directorias. Indique claramente o que representa cada uma das variáveis que utilizar e explicite a chamada inicial.